

BREAKING NEWS

***What media says
about AI***





Sofia Ginalis

*Portugal
Correspondent*



Marvin Löhlein

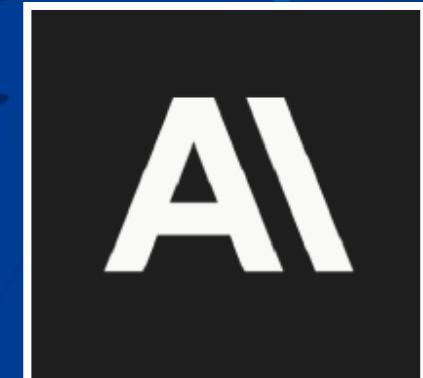
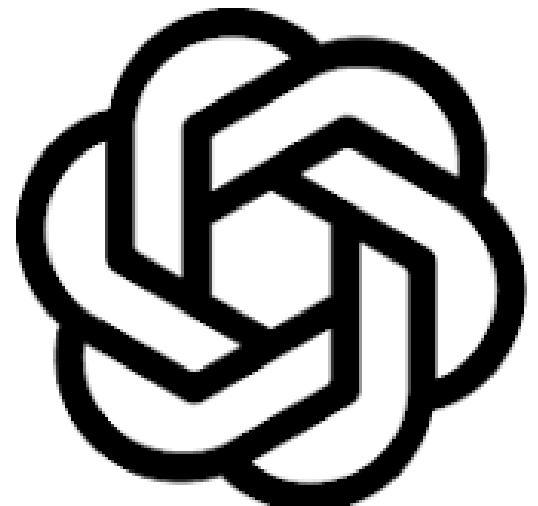
*Thailand
Correspondent*

LIVE

REPORTERS

Das Thema

- **Kontext:** KI in 2024 vom Hype zur Anwendung
- **Konflikt:** Kampf um Marktführerschaft zwischen Google (Gemini), OpenAI (GPT-4o) und Anthropic (Claude, Perplexity)
- **Frage:**
 - Wie berichten Medien über KI Unternehmen und Modelle?



Der Zeitraum

12.02.2024 - 19.02.2024

Google

- Gemini 1.5 Pro:
 - Context Window von 1 Million Token

OpenAI

- Sora announcement

13.05.2024 - 20.05.2024

OpenAI

- GPT-4o ("Omni")
- Voice Modell "Sky"
- Auflösung Super Alignment Team

Google

- veo announcement



Datenquellen

1. GDELT (Der Kompass)

- **Was:** Globale Datenbank für Ereignisse & News
- **Vorteil:** Erkennt Trends & Sentiment
- **Nutzen:** Filtert Relevanz aus dem Chaos

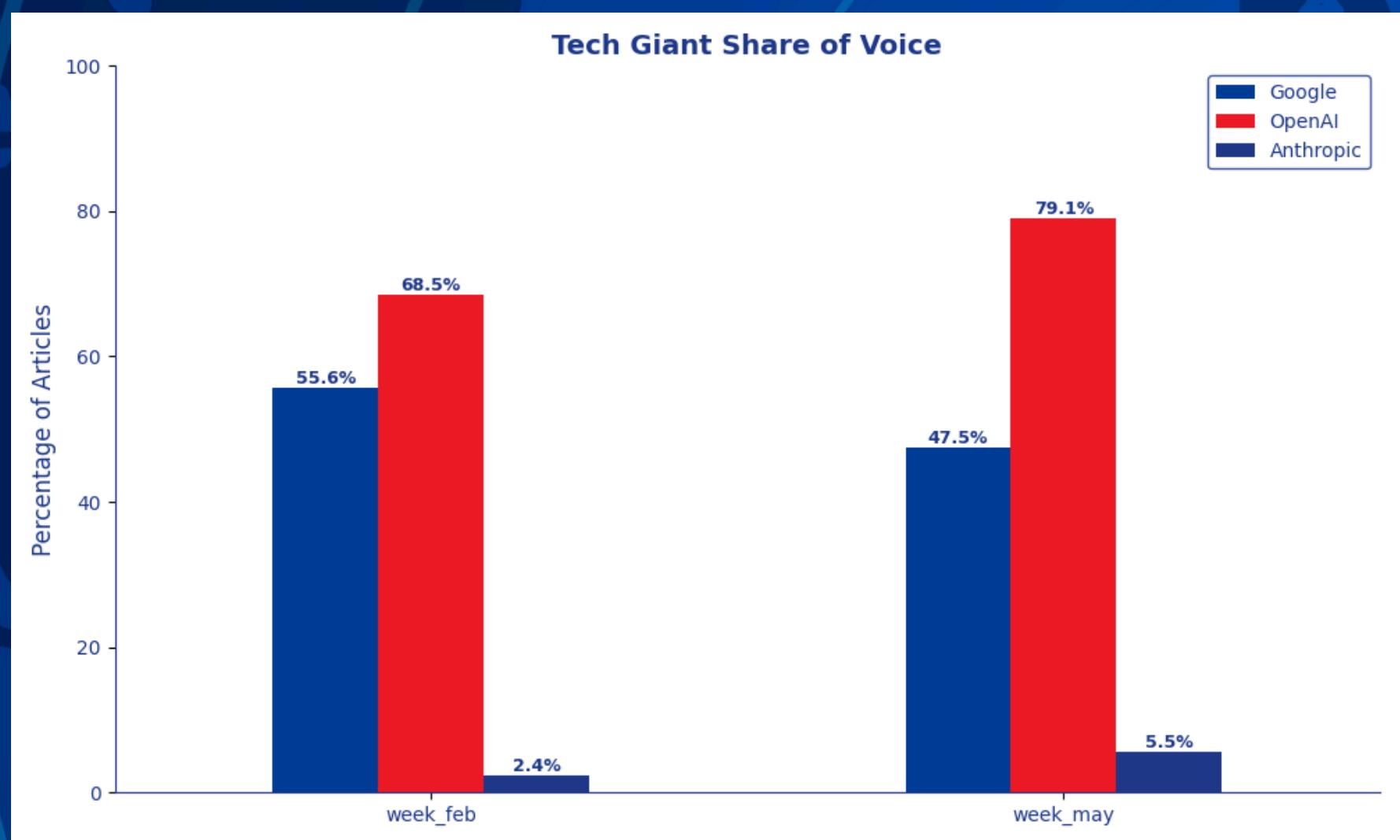
The GDELT Project

2. Common Crawl (Die Bibliothek)

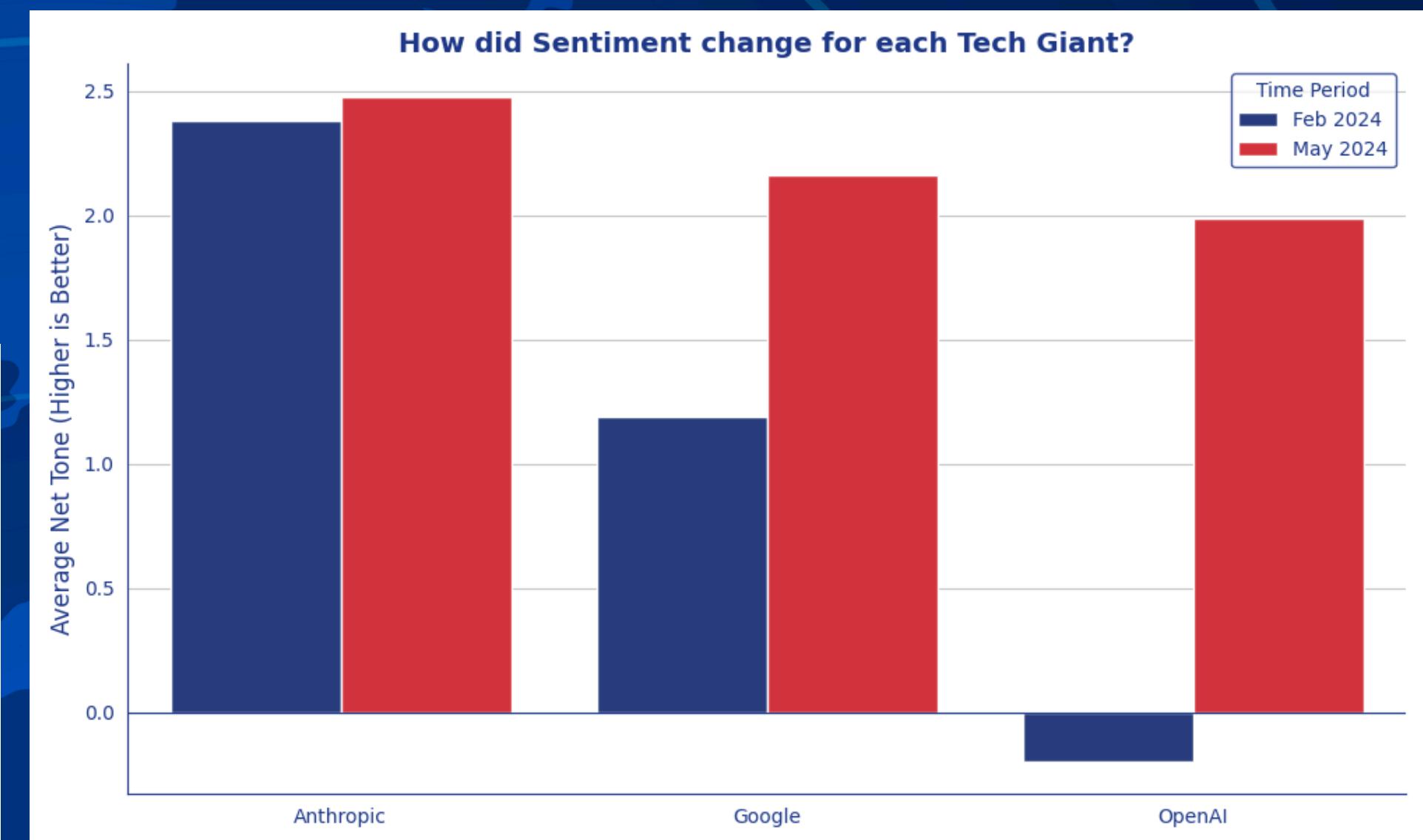
- **Was:** Riesiges, offenes Web-Archiv
- **Vorteil:** Zugriff auf kompletten Volltext
- **Nutzen:** Basis für detaillierte Textanalyse

Common Crawl



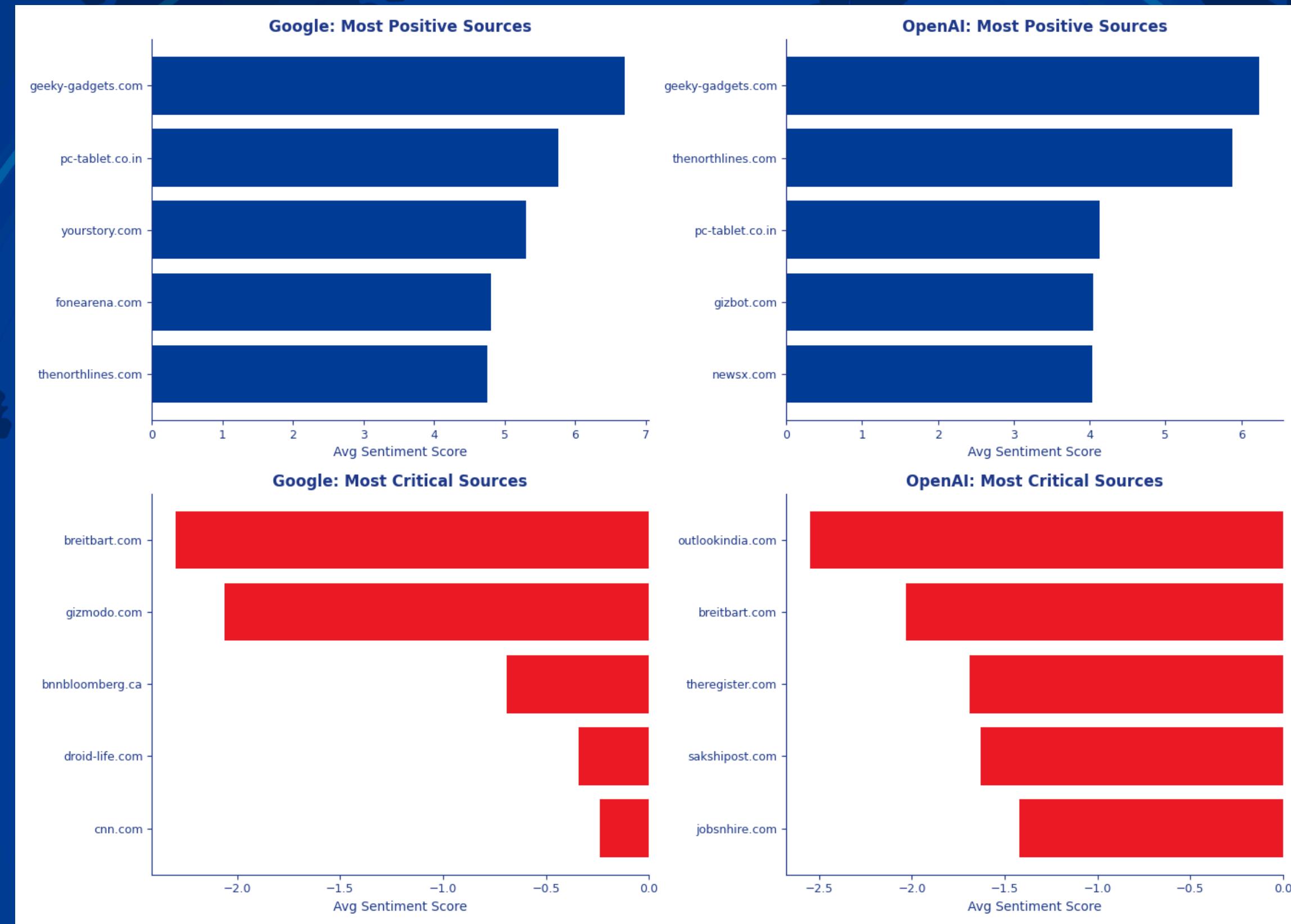


Share of Voice



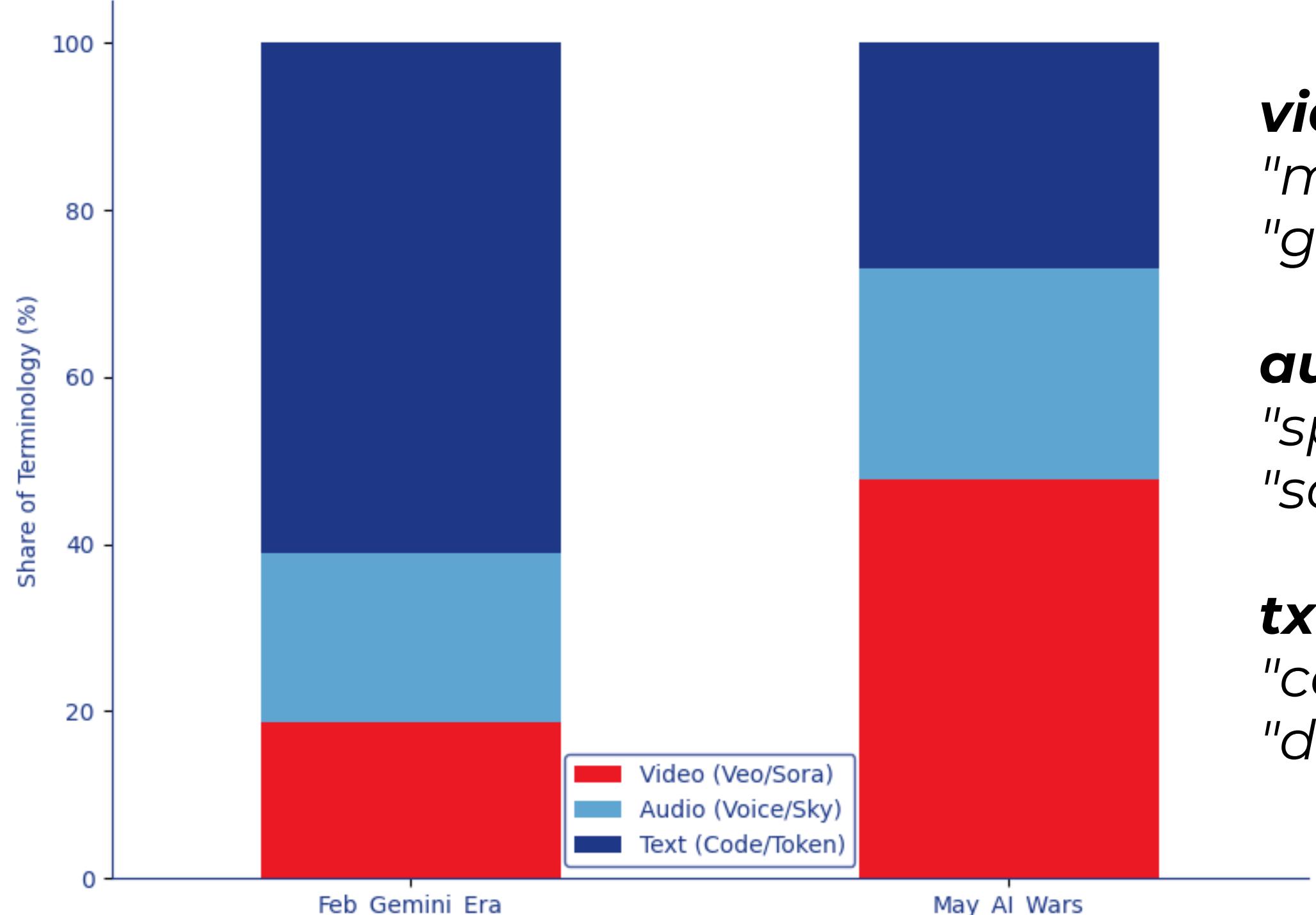
Sentiment

FRIENDS VS. CRITICS



Keyword frequency counting

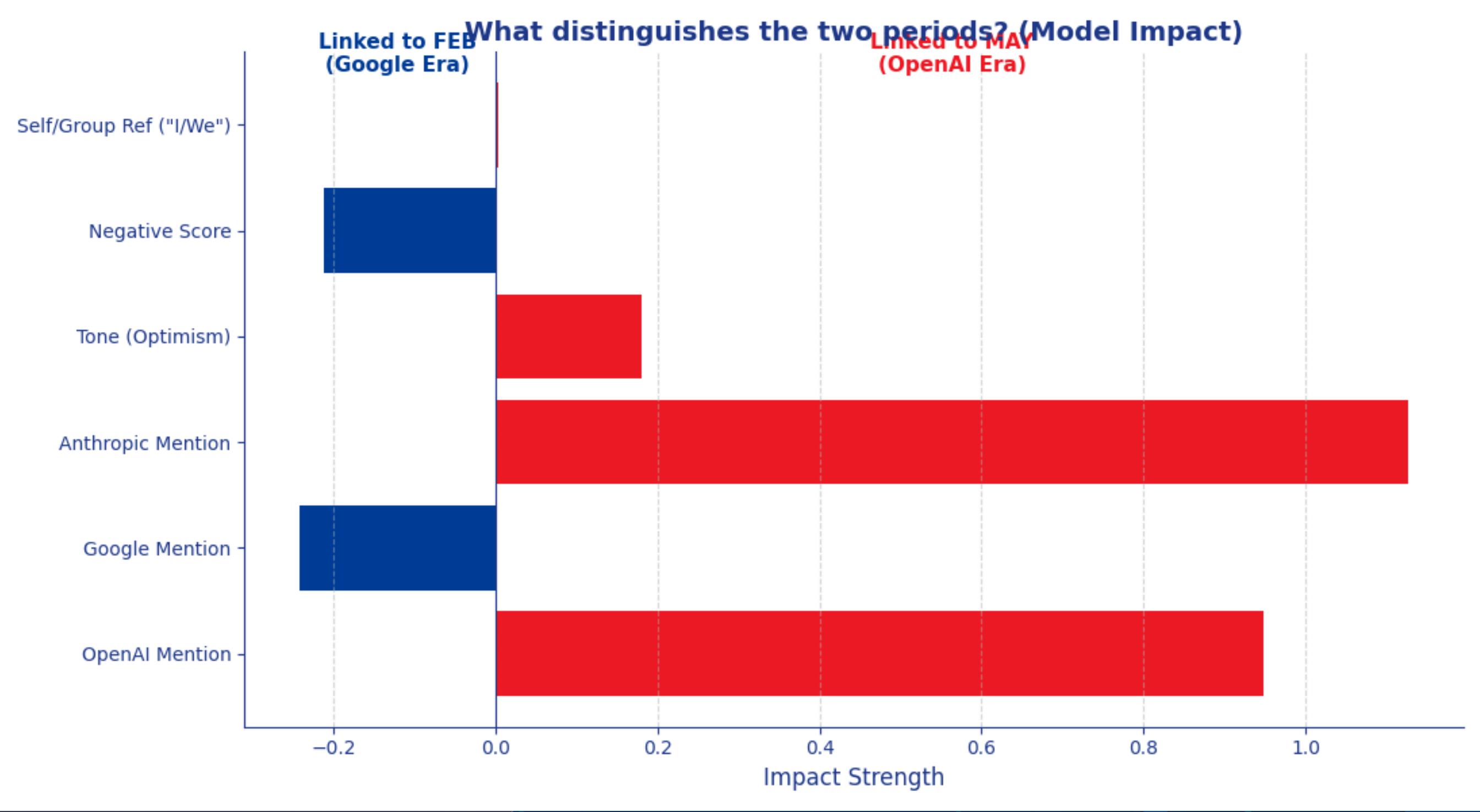
The Modality Shift: From Text (Feb) to Video/Voice (May)



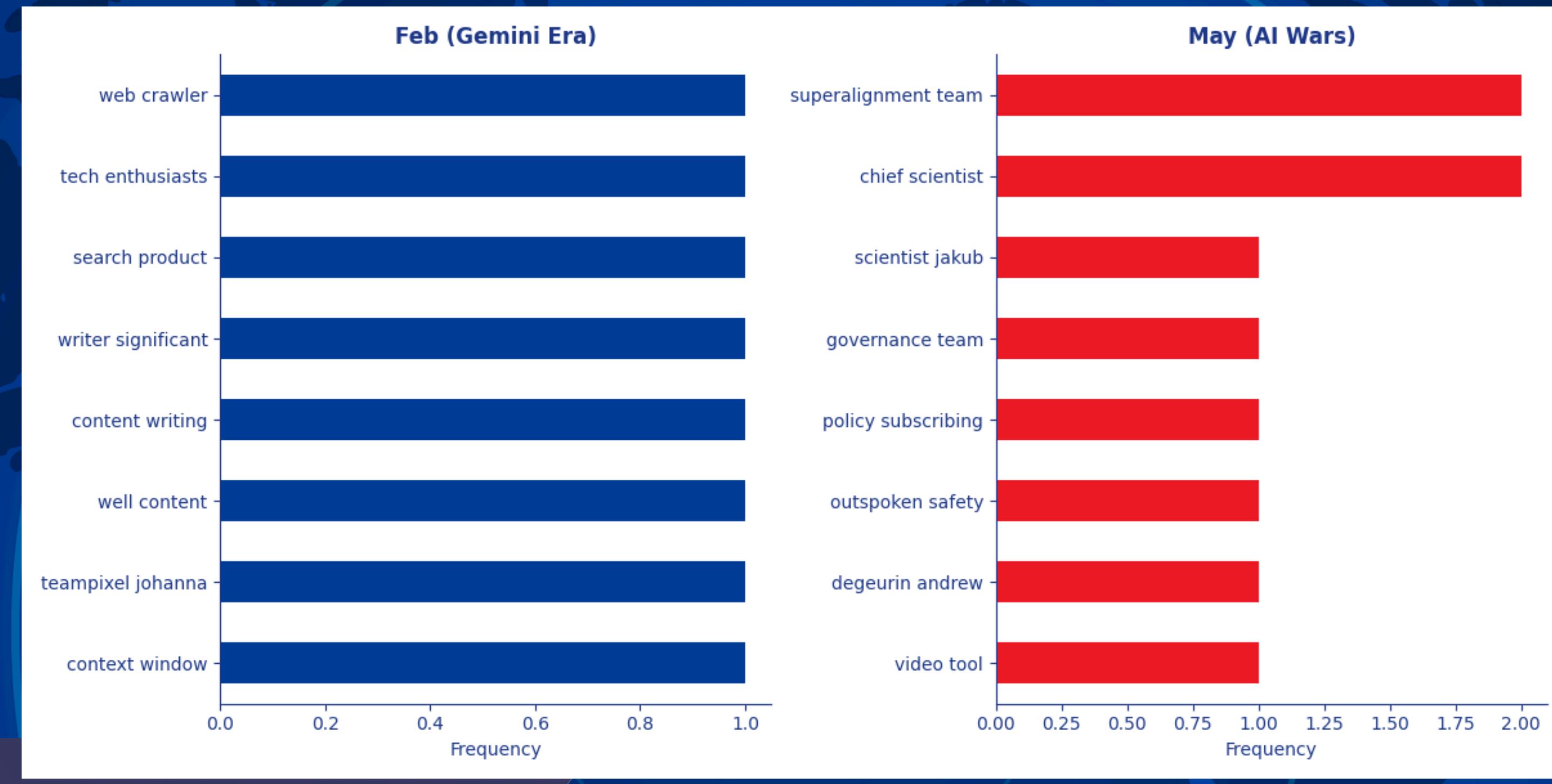
vid_words = ["video", "sora", "veo",
"movie", "film", "camera",
"generation"]

aud_words = ["voice", "audio",
"speech", "listen", "talk", "hear",
"scarlett", "sky"]

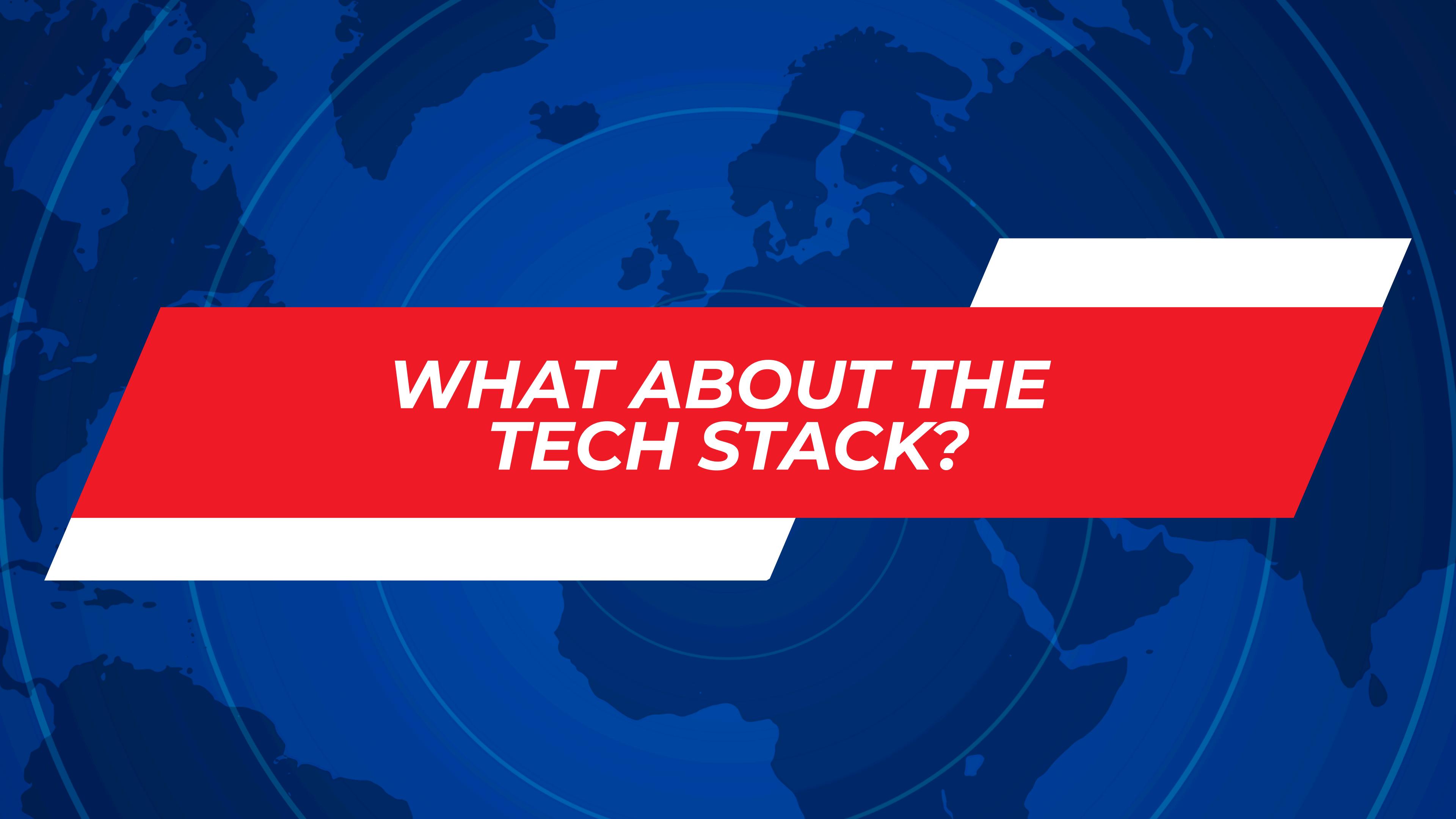
txt_words = ["text", "code", "token",
"context", "read", "summary",
"document"]



MACHINE LEARNING: FEATURE IMPORTANCE



MACHINE LEARNING: NLP ANALYSIS



**WHAT ABOUT THE
TECH STACK?**

Data Storage Layer: AWS Infrastruktur & Design-Entscheidungen

Provider-Wahl & Kostenmanagement

- **Plattform:** Amazon Web Services (AWS).
- **Kostenoptimierung:** Verzicht auf permanente EC2-Instanzen; stattdessen Nutzung von lokaler Rechenleistung und On-Demand Services.

Speicherstrategie (Data Lake)

- **Storage:** Amazon S3 als zentraler Datenspeicher (Data Lake) für Rohdaten und prozessierte Daten.
- **Struktur:** Klassische Ordnerstruktur (Filesystem) statt relationaler Datenbank (RDBMS), da für Big Data und unstrukturierte JSON-Daten (GDELT/Common Crawl) besser geeignet.



Data Source Layer: GDELT Project (Global Database of Events, Language, and Tone)

- **Ziel:** Identifikation relevanter Artikel-URLs über KI.
- **Alter Ansatz:** Querying von täglichen Daten aus 2024 von GDELT Articles gefetched
- **Neuer Ansatz:** Relevante Daten auf zwei Wochen beschränken und von BigQuery bekommen
- **Filter-Kriterien:**
 - **Keywords:** OpenAI, Google, Anthropic + generell KI.
 - **Zeitraum:**
 - 12.02.2024 - 19.02.2024 (Gemini 1.5, Sora)
 - 13.05.2024 - 20.05.2024 (GPT-4o, Google conference)

The GDELT Project

Data Storage Layer: Common Crawl (Web Archive)

- **Ziel:** Beschaffung des Volltext-HTMLs zu den URLs, die via GDELT gefunden wurden.
- **Funktionsweise:** CC speichert Snapshots des Internets in riesigen Batches (ca. 10 pro Jahr).
- **Workflow (Spark auf AWS Glue):**
 1. **Ingestion:** Laden der GDELT-JSONs (URLs) vom S3 in einen Spark DataFrame.
 2. **Index Lookup:** Für jede URL muss der korrekte Common Crawl Index durchsucht werden, um die physikalische Adresse der Daten im Batch zu finden.
 3. **Extraction:** Gezielter Download der HTML-Daten (kein Download ganzer TB-Batches).
 4. **Storage:** Speicherung der rohen HTML-Files im S3 Bucket.

Common Crawl



Data Processing & Feature Engineering (GDELT)

GDELT

- **Parsing & Strukturierung:**
 - Normalisieren von Zeitstempeln + Extraktion der Root-Domains.
 - Themes und Organizations aufsplitten in analysierbare Arrays.
- **Deduplizierung:** Bereinigung auf eindeutige Zeile pro URL.
- **Zeit-Labeling:** Zuweisung der Zeitfenster für Vergleichsanalysen.

Common Crawl

- **Bereinigen & Normalisieren:**
 - Entfernung von HTML-Tags und Sonderzeichen; Umwandlung in Kleinschreibung.
- **Token-Filterung:**
 - Benutzerdefinierte Stoppwörter sowie Tokens mit weniger als 3 Zeichen ausfiltern.

Output & Analytics Layer: Spark Analysis

1. Entity Sentiment Analysis (SQL/Spark Analytics)

- **Ansatz:** Aggregation mittels SQL GROUP BY und AVG.
- **Query:** Berechnung des durchschnittlichen Tons (AvgTone) pro Unternehmen und Woche unter Verwendung der Flags k_google, k_openai und k_anthropic.
- **Output:** Vergleichstabelle der Sentiment-Veränderungen zwischen den Zeiträumen sowie grafische Visualisierung mittels Matplotlib.

2. Network Analysis

- **Approach:** Spark SQL + explode() for array columns
- **Analysis:** Top news sources by article count, co-occurring organizations (exploded from orgs_arr)
- **Output:** Publisher rankings + corporate ecosystem maps (who appears alongside whom)

Output & Analytics Layer: Spark Analysis + ML GDELT

3. Logistische Regression (Primärmodell)

- **Zweck:** Binäre Klassifikation der Zeiträume (Februar vs. Mai).
- **Features:** Unternehmens-Erwähnungen (OpenAI/Google/Anthropic) und Tone-Metriken (Sentiment, Negativität, Polarität, Dichte der Selbstreferenzen).
- **Performance:** AUC ~0.72
 - **72%** chance that our model will successfully distinguish between articles February and May

Output & Analytics Layer: NLP of Common Crawl Files

Challenge: Nur ~50 HTML-Samples verfügbar (Probleme beim Download) → Ergebnisse nicht statistisch signifikant

Text-Verarbeitung:

- **Stopword-Removal:** Erweiterte Liste (Standard + "click", "share", "advertisement", "google", "openai", "gemini", "gpt")
- **Token-Filter:** Nur Wörter >3 Zeichen (eliminiert HTML-Reste)

Feature Extraction:

- **Bigram-Generierung (N-Gram n=2):** Phrase-Level-Analyse ("context window", "voice mode")
- **CountVectorizer:** Vokabular von 1500 häufigsten Bigrammen
- **TF-IDF:** Gewichtung nach Relevanz (häufig im Dokument, selten im Korpus)

•



CHALLENGES AND LEARNINGS

Challenges

- **Ursprungsidee nicht praktikabel**
- **Common Crawl:**
 - Massive Datenmengen (Terabytes)
 - Datensuche mittels Indexen extrem zeitaufwendig
 - Serverüberlastung: Ende Dezember häufig Verbindungsabbrüche
- **Datenkomplexität:**
 - Rohdaten mussten aufwendig bereinigt und transformiert werden, um "ML-ready" zu sein

Learnings

- **Planung und Kommunikation**
- **Infrastruktur & Kosten:**
 - **Serverlos besser:** AWS Glue & S3 waren viel günstiger und wartungsärmer als dauerhafte Server (EC2).
- **Keine Datenbank nötig:** Für Big-Data-Analysen reicht eine saubere Ordnerstruktur (Parquet/JSON) völlig aus
- **Strategieanpassung:**
 - **Radikale Eingrenzung:** Statt "alles" zu wollen, müssen Zeitfenster (z.B. nur Feb/Mai) und Themen extrem scharf definiert werden.

Learnings

- **Intelligente Suche:** Gezielte Index-Abfragen (Wahrscheinlichkeits-Ansatz) sind effektiver als chronologisches Abarbeiten massiver Datenmengen.
- **Prozess-Optimierung:**
 - **Realismus beim Cleaning:** Rohdaten sind niemals "ML-ready": der Pipeline-Bau (Parsing -> Labeling -> Engineering) frisst viel Zeit.
 - **Dokumentation:** Ein "Data Dictionary" muss zu Beginn stehen, damit das Team Datenstrukturen einheitlich versteht.



ANY QUESTIONS?



GitHub Repo

<https://github.com/JuliaPabst/Big-Data-AI-News-Analysis>